Rishigesh J Jayananth

rjayanan@ucsd.edu | (916) 932- 6936|

Education

University of California, San Diego San Diego, CA
BS/MS - June 2025
Master of Science in Data Science and Machine Learning 2025
Bachelor of Science in Computer Engineering 2024
Mathematics Minor
GPA: 3.5/4.0

Mira Loma High School Sacramento, CA
June 2020
High School Diploma
IB Diploma
Weighted GPA: 4.6
Valedictorian Medal

Coursework

- Artificial Intelligence: Search and Reasoning +Statistical Learning I (Generative Learning)
- Introduction to Machine Learning + Linear Algebra + Prob & Stats for Data Science
- VLSI for Machine Learning + GPU Programming
- Optimization and Acceleration of Deep Learning on Various Hardware Platforms
- Introduction to Stochastic Process + Introduction to Computational Stochastics + Random Process
- Advanced Digital Design + Components and Design Techniques for Digital Systems
- Operating Systems + Computer Architecture
- Computer Organization and Systems Programming + Digital Systems Lab
- Advanced Data Structures + Design and Analysis of Algorithms
- Software Engineering + Software Labs and Techniques

Skills

Programming Languages: Python, Javascript, HTML/CSS, Java , C++,C, ARM, Cudas
Software and Tools: VS Code, Git/Bitbucket, Valgrind, GDB/JDB, VIM, Virtual Env, Linux, WSL, X2Go, Putty, Docker, Kubernetes
Operating Systems: Microsoft Windows OS, Unix/Linux
Hardware Skills: Oscillator, Power Generator, Circuit Analysis, SystemVerilog, RLT Design, GPU, Parallel Programming
Other Technologies/Tools: Confluence, MS teams, Jira

Experience

Hewlett Packard Enterprise (Aruba Networks) - System Validation Engineer Intern June 2022- Dec 2022

- Developed a GUI for an Aruba Product(VC Sim) that simulates switches and access points - Python
  - Programming Techniques/Libraries used: Tkinter, NCurses, Rest API, OAuth, Yaml Files
  - Backend Technologies/Libraries used: Multithreading, Gevent, Monkey Patch, OS module
  - Other Technologies Used: Confluence, MS Teams, Jira, k8
- System Testing Automation - Python
  - Executed manual testing before building automation for various system test cases to ensure each step of the automated test was understood
  - Completed automated test cases would get added to a Test Suite and run weekly
  - Programming Techniques/Libraries used: Selenium, Pytest, Rest API

Solecta - IT Tech Intern July 2023- September 2023

- Push features in Production Database to help automate processes used in operation pipeline in manufacturing products
- Validate new features in Production Database used in Operation and Operation Management
- Collaborated with IT team and Operations team to build this solution used in the production database
  - Softwares used: SQL, MS Access, MS Excel, VBA, Power BI

Projects

Frugal - A Simple Budgeting App - Front End Lead

- An app built natively with Html/CSS + Javascript using local Storage to help users budget their finances
- Managed and facilitated the front end team which consisted of two front end developers and graphic designer
- Developed the front end for the login page and home page
- Built the backend functionality of filtering and searching expenses
- Helped build the local Storage for storing expenses
- Tested and built a new pipeline for html validation through Github Actions
- Helped testing team write E2E test cases to test home page,login page, and create account page functionality
- Documented meeting minutes and kept a list of ongoing tasks to keep accountability of self and teammates

NN hardware on FPGA - Ongoing

- Wrote Python Script to generate weights and bias txt for hardware to use as ROM
- Built necessary hardware components in System Verilog to be synthesize to perform inference with the given weights and biases for each layer in Neural Network Model
- WIP: Write testbench on top level module where hardware is synthesizes for verification by matching predicted output on hardware with the predicted output on software

Scalable Designs of CNN on 2D Systolic Array with Pruning and Compression

- Software- Implemented VGG model and changed model architecture on layer 27 to input channel and output channel to 8 so hardware can later replicate
- Used 2D Systolic Architecture to have 64 Processing Elements to handle 8 input channels and 8 output channels
- Extension 1: Scalability - By adding multichannel PE and multicore, hardware can handle various amounts of input channels and output channels, being able to scale model on a specific layer horizontally and vertically
- Extension 2: Power Efficient - By adding structured pruning and Huffman compression, our hardware was able to utilize less power for computation with a small drawback of lower accuracy - by adding a hardware component of mac gating, we can save power on consumption with weights of 0